

ROS Basics

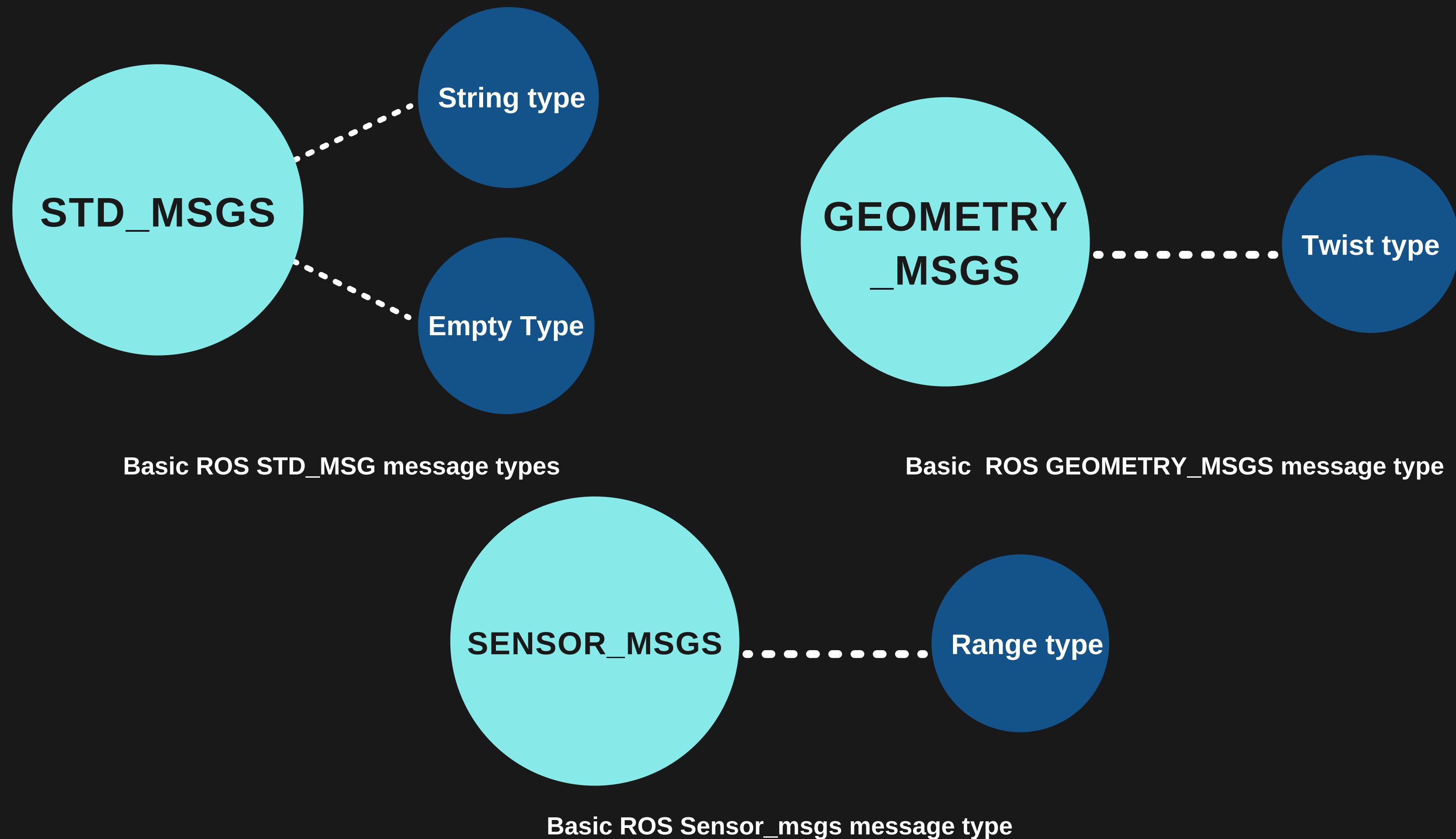
Topic Covered: ROS
std_msgs,geometry_msgs,sensor_msgs

Part 3

100Days of ROS series

By
M.N.Subhash
@malladisubhash

DIFFERENT ROS MESSAGES



STD_MSG ROS MESSAGE PACKAGE

The two standard ROS message types implemented as a part of the task are:

- String ros message type
- Empty ros message type

String ROS messages:

1. The definition of the string message types is "string data", which means it accepts the data in the string format
2. The snippets from the code are :

```
char hello[13] = "hello world!"; // This is the message that will be published to the channel chatter  
  
// The message "hello" is assigned to definition "data" and the message is published to the in the channel  
chatter  
str_msg.data = hello;  
chatter.publish( &str_msg );
```

Empty ROS messages :

1. The definition of the empty message type is **nothing**, its just a trigger and can be used to send a reaction to a event
2. The snippets from the code are:

```
//This is the callback function  
void messageCb( const std_msgs::Empty& toggle_msg){  
digitalWrite(LED_BUILTIN, HIGH-digitalRead(LED_BUILTIN));  
}
```

```
//The subscriber listening to the channel, will trigger the event when the empty message is published  
ros::Subscriber<std_msgs::Empty> sub("toggle_led", &messageCb);  
// "toggle_led" is the channel through which message is published by the publisher and subscribed by  
subscriber "sub"
```

This explains the important snippets of the code. Please go through the GitHub for the complete code
link: https://github.com/malladi2610/100_days_of_ROS/tree/master/Day%202

GEOMETRY_MSGS ROS MESSAGE PACKAGE

Geometry_msgs provides messages for common geometric primitives such as points, vectors, and poses.

The ROS message types implemented as a part of the task are:

- **Twist ROS message:**

1. The definition of the twist ROS message type is linear and angular, both of these are vector datatypes and contains data about x,y and z axis and the angular message about roll,pitch and yaw.
2. The snippets from the code of task 5 are:

The complete code is available in : https://github.com/malladi2610/100_days_of_ROS/tree/master/Day%204%20-%2010/Task%205

In this task Arduino acts a publisher and publishes the joystick values to the Master PC and this controls the turtle in the turtle sim,

```
// Here the "twist" is the publisher name and the channel name is "twist_msgs"  
geometry_msgs::Twist twist_msgs;  
ros::Publisher twist("/turtle1/cmd_vel",&twist_msgs);
```

// Here the Linear type data is sent through the channel and the x and y directions are preferred as the turtle sim is a 2-D simulator

//Right direction

```
if(mapX == -512 && mapX >= -512){  
    xpos = xpos + 1;  
    twist_msgs.linear.x = xpos;  
    twist_msgs.linear.y = 0;  
}
```

//Left direction

```
if(mapX == 512 && mapX <= 512){  
    xpos = xpos - 1;  
    twist_msgs.linear.x = xpos;  
    twist_msgs.linear.y = 0;  
}
```

//Upward direction

```
if(mapY == -512 && mapY >= -512){  
    ypos = ypos + 1;  
    twist_msgs.linear.x = 0;  
    twist_msgs.linear.y = ypos;  
}
```

//Downward direction

```
if(mapY == 512 && mapY <= 512){  
    ypos = ypos - 1;  
    twist_msgs.linear.x = 0;  
    twist_msgs.linear.y = ypos;  
}
```

} // This is the snippet of the code which is responsible for the movement of the robot in x and y direction

About Me

I am currently pursuing my degree at Sir M Visvesvaraya Institute of Technology, Bangalore.

The domains I am interested in are Electronics and Robotics.

My interest in Robotics started when I was in my second year, Currently improving my skills in ROS and took a 100daysofROS challenge to focus mainly on the hardware implementation of ROS

I put regular updates of my progression of challenges on Twitter
As a part of the challenge, I will be documenting the important things and will be making a concise version of them.

Follow me to catch up with ROS and learn how to implement it on hardware.

LinkedIn profile: linkedin.com/in/malladisubhash

Twitter Handle: [@mns2610](https://twitter.com/@mns2610)

GitHub ROS repository link: https://github.com/malladi2610/100_days_of_ROS